

КОНЦЕПЦИЯ МНОГОЗАДАЧНОЙ ГРИД-СИСТЕМЫ С ГИБКИМ РАСПРЕДЕЛЕНИЕМ СВОБОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ СУПЕРКОМПЬЮТЕРОВ¹

Афанасьев А.П., Бычков И.В., Заикин О.С.,
Манзюк М.О. Посыпкин М.А., Семенов А.А.

Институт динамики систем и теории управления СО РАН, Иркутск
Институт проблем передачи информации им. А.А. Харкевича РАН, Москва

Работа посвящена анализу двух проблем современных высокопроизводительных вычислений. Первая проблема состоит в наличии востребованных практикой задач, для решения которых требуются сверхбольшие вычислительные ресурсы. Вторая заключается в систематической недозагруженности суперкомпьютеров. В статье в качестве пути решения данных проблем мы предлагаем концепцию многозадачной грид-системы с гибким распределением свободных вычислительных ресурсов суперкомпьютеров. Базовые принципы такой системы состоят в том, что она использует только свободные ресурсы суперкомпьютеров и делает это только на правах обычного пользователя, тем самым никак не конфликтуя с политикой администрирования. Мы демонстрируем работоспособность предложенной нами концепции на примере CluBORun-грид – реально функционирующей грид-системы, которая на текущий момент использует свободные ресурсы суперкомпьютеров для усиления вычислительной мощности проектов добровольных вычислений SAT@home и OPTIMA@home.

1. Введение

В настоящее время существует обширный класс задач, в отношении которых возможно довольно точно оценить объем вычислительных ресурсов, требуемых для их решения. Соответствующее значение может оказаться очень большим (сотни эксафлоп). Тем самым решение такой задачи даже на очень мощном суперкомпьютере может потребовать продолжительного времени (месяцы и даже годы). Получить на такое время суперкомпьютер для одной задачи, как правило, нереально, за исключением ситуаций, когда суперкомпьютер создан специально «под конкретную проблему» (например, в ядерных центрах).

В то же время, одна из важнейших проблем, возникающих при работе с современными суперкомпьютерами, состоит в их неравномерной

¹Работа выполнена при частичной финансовой поддержке РФФИ (гранты №№ 14-07-00403, 13-07-00768). Совета по грантам Президента РФ для поддержки молодых ученых (стипендия СП-1855.2012.5), Совета по грантам Президента РФ для государственной поддержки ведущих научных школ (НШ-5007.2014.9).

загруженности. В одни моменты суперкомпьютер может быть загружен более чем на 90%, а в другие – менее чем наполовину. Это очень серьезная проблема, поскольку даже при значительной недогрузке суперкомпьютер продолжает потреблять существенный объем электроэнергии. Мониторинг данной проблемы по нескольким кластерам показывает, что средняя загруженность за год лишь для отдельных кластеров может быть выше 70%, составляя обычно около 50% (а иногда и существенно меньше).

Сказанное выше выглядит парадоксально: с одной стороны, существуют востребованные практикой задачи, для решения которых не подходят современные политики использования суперкомпьютеров, с другой – значительную часть своего времени суперкомпьютеры не загружены на полную мощность, что приводит к удорожанию вычислений.

Проанализировав перечисленные проблемы в комплексе, мы предлагаем, как один из возможных путей их решения, многозадачную грид-систему с гибким управлением свободными вычислительными ресурсами суперкомпьютеров (далее «кластеров»). Концептуально предлагаемая грид-система основана на идеях, используемых в современных проектах добровольных распределенных вычислений (ДРВ). Также мы описываем CluBORun-грид – первый прототип такой системы, который на текущий момент используется для привлечения свободных ресурсов кластеров в проекты ДРВ с целью увеличения вычислительной мощности последних. Данная грид-система прошла первые успешные испытания на решении ряда крупномасштабных вычислительных задач в проектах ДРВ SAT@home и OPTIMA@home.

2. Добровольные распределенные вычисления и концепция многозадачной грид с гибким распределением свободных ресурсов суперкомпьютеров

Начнем с краткого описания основных принципов работы проектов ДРВ. В рамках идеологии добровольных вычислений вычислительные ресурсы предоставляются частными лицами, которые называются добровольцами. Как правило, добровольцы предоставляют ресурсы собственных ПК. Для проектов ДРВ подходят задачи, решение которых может быть получено посредством разбиения на подзадачи меньшей размерности с последующей независимой их обработкой (т.н. Coarse Grained Parallelism). Подзадачи, получаемые в результате разбиения исходной задачи, далее называем элементарными. Разбиение должно быть сделано так, чтобы на решение каждой элементарной задачи даже на слабом вычислительном узле (в проектах ДРВ принят термин «хост») уходило относительно немного времени – как правило, от нескольких минут, до нескольких часов.

Существуют специальные средства организации проектов ДРВ. Наиболее известной в этом плане является платформа BOINC [1]. Ниже перечислены базовые принципы BOINC-вычислений.

1. Управление работой проекта осуществляет центр администрирования. Данный центр включает выделенный сервер, работающий в круглосуточном режиме. Функции центра администрирования состоят в следующем: создание клиентских приложений, генерация пакетов элементарных заданий и их рассылка участникам проекта, сбор и обработка результатов, присылаемых участниками проекта. В рамках одного центра возможно управление несколькими BOINC-проектами.
2. Ресурсы проекта состоят из хостов участников проекта, называемых добровольцами. Каждый доброволец получает с сервера проекта клиентское приложение и пакеты элементарных заданий. Результаты их обработки отправляются на сервер проекта. При решении каждой элементарной задачи задействуются только свободные ресурсы хоста. Управление ресурсами на хосте осуществляет специальная программа, называемая BOINC-менеджером. Возможности BOINC-менеджера подразумевают весьма гибкие настройки используемых ресурсов, преследуя цель не мешать работе пользователя. Можно настроить BOINC-менеджер таким образом, что соответствующие задачи будут решаться лишь в те временные промежутки, когда пользователь никак не задействует хост. Промежуточные результаты вычислений должны сохраняться. Так, если решение элементарной задачи было прервано из-за того что пользователю потребовались ресурсы, BOINC-менеджер должен иметь возможность вернуться к процессу решения этой задачи желательно с той точки, в которой предыдущее вычисление было прервано.
3. На одном хосте возможно решение нескольких различных задач – BOINC-менеджер делит свободные ресурсы хоста в соответствии с задаваемыми приоритетами.

Перечисленные принципы мы использовали для формулировки концепции грид-системы, способной гибко распределять свободные вычислительные ресурсы кластеров между несколькими задачами. Основные принципы данной концепции состоят в следующем.

- i. Наличие центра администрирования грид (аналог центра администрирования добровольного проекта). Данный компонент отвечает за разработку клиентских приложений, построение пакетов элементарных заданий, их рассылку по кластерам и обработку получаемых от кластеров результатов.
- ii. Использование вычислительных узлов кластеров (аналоги хостов добровольцев). Основные требования: грид-система должна

задействовать для решения своих задач только свободные ресурсы кластера; грид-система должна обращаться к кластеру только с правами стандартного пользователя и, тем самым, никак не конфликтовать с политикой администрирования данного кластера.

- iii. На каждом кластере должно быть возможным решение нескольких различных задач. Свободные ресурсы кластера при этом делятся между задачами в соответствии с приоритетами, определяемыми администраторами грид-системы.

Важнейший специфический момент грид-системы, удовлетворяющей перечисленным требованиям, состоит в необходимости механизмов взаимодействия с кластером в плане запросов, использования и освобождения вычислительных ресурсов (в VOINC-проектах соответствующие функции реализуются через стандартные службы VOINC-менеджера).

Следует отметить, что на каждом кластере имеется система управления очередью заданий, которая делит свободные ресурсы кластера между задачами пользователей. В рамках предлагаемой нами концепции взаимодействие с очередью заданий кластера с целью использования его свободных вычислительных ресурсов должно быть реализовано в виде полностью автоматической системы, которую далее мы называем СИСП (система использования свободных ресурсов). Такая система должна удовлетворять целому ряду требований. Особо подчеркнем, что политика управления очередью заданий определяется администратором кластера. Мы исходим из того, что СИСП не должна никак вмешиваться в эту политику. Иными словами, в своей работе такая система должна задействовать только стандартные пользовательские права. Для этого СИСП должна осуществлять мониторинг очереди заданий и занимать только свободные ресурсы под видом обычного пользователя. В процессе мониторинга очереди СИСП должна учитывать большое число нюансов. Например, даже при наличии значительного числа свободных ресурсов СИСП не должна занимать их на продолжительное время, поскольку в противном случае возможен «эффект испуга» зашедшего в последующий момент на кластер пользователя. Если же СИСП берет свободные ресурсы на малое время (от нескольких минут до нескольких часов), и соответствующая информация отображается в очереди, таких ситуаций возникать не должно – пользователь ставит свою задачу в очередь, понимая, что через малое время значительные вычислительные ресурсы будут освобождены. Если СИСП ставит задания, например, на несколько часов, и в очереди появляются задания других пользователей, то СИСП должна уметь снимать свои задания, высвобождая вычислительные ресурсы, и сохранять результаты промежуточных вычислений.

3. CluBORun-грид – многозадачная грид-система с гибким распределением свободных ресурсов кластеров

В данном пункте мы демонстрируем возможность достижения поставленных выше целей на примере CluBORun-грид – разработанной нами и реально функционирующей многозадачной грид с гибким управлением свободными ресурсами кластеров. На данном этапе CluBORun-грид используется для усиления вычислительной мощности проектов ДРВ SAT@home и OPTIMA@home. CluBORun – это сокращение от Cluster for BOINC Run. Далее мы описываем основные компоненты данной системы.

Конечной целью работы CluBORun является запуск клиентских приложений BOINC-проектов на вычислительных ресурсах кластера посредством BOINC-менеджера. Использование BOINC-менеджера обеспечивает фактор многозадачности – на свободных ядрах могут считаться одновременно несколько задач из различных проектов. Следует отметить, что стандартно задачи на кластере запускаются в виде MPI-приложений, тогда как BOINC-менеджер таковым не является. С другой стороны, BOINC-менеджер обладает собственными функциями захвата и освобождения любого количества ресурсов вычислительного узла, на котором он запущен (как правило, хоста проекта ДРВ). Для корректного запуска BOINC-менеджера посредством стандартного взаимодействия с очередью кластера мы использовали следующий подход. Была создана специальная MPI-программа, названная start_boinc, которая запускается на кластере стандартным путем (через вызов mpiexec).

Опишем работу start_boinc на произвольном узле кластера. Предположим, что рассматриваемый узел имеет n процессорных ядер (везде речь идет только о CPU-ядрах). Поскольку программа start_boinc является стандартным MPI-приложением, средствами MPI-среды кластера ей выделяется для работы $\leq n$ MPI-процессов. Далее для простоты считаем, что start_boinc после запуска на узле кластера получает n MPI-процессов. На одном MPI-процессе, который далее называется управляющим, start_boinc запускает BOINC-менеджер, оставшиеся $n - 1$ процессов программой start_boinc не задействуются. Для этих MPI-процессов мы используем термин «спящие». После запуска BOINC-менеджер «видит» все свободные ресурсы узла: как соответствующие спящим MPI-процессам, так и тому MPI-процессу, на котором он запущен. После этого BOINC-менеджер задействует все свободные ресурсы узла так, как если бы это были свободные ресурсы хоста проекта ДРВ. Таким образом, на одном узле кластера могут вестись вычисления для различных проектов ДРВ. При необходимости завершения расчетов программа start_boinc подает BOINC-менеджеру соответствующую команду, которая отслеживается и обрабатывается собственными службами BOINC-менеджера.

Как мы уже отмечали выше, мониторинг и захват свободных ресурсов кластера в CluBORun-грид осуществляется с правами обычного пользователя и никак не конфликтует с политикой администрирования кластера. Эти функции выполняет СИСР CluBORun, работа которой организована в форме взаимодействия двух скриптов – один из которых отвечает за мониторинг ресурсов кластера, а другой – за запуск на свободных узлах программы start_boinc.

Структурно СИСР CluBORun состоит из следующих компонентов.

1. Каталоги, соответствующие MPI-задачам start_boinc, которые должны быть решены на кластере (функции каждой такой задачи состоят в запуске одного или нескольких экземпляров BOINC-менеджера).

2. Файлы-флаги, соответствующие MPI-задачам start_boinc: в случае возможности запустить конкретную MPI-задачу создается start-флаг, а при необходимости ее завершить создается stop-флаг.

3. Текстовый файл all_tasks.txt – перечень запускаемых MPI-задач. Для каждой задачи в данном файле указывается имя start-флага, имя stop-флага, путь к каталогу этой задачи. Пример содержимого файла all_tasks.txt с описанием двух задач:

```
task001 task001.start task001.stop /home/user/SAT node001 node002 ... node008
task002 task002.start task002.stop /home/user/SAT node009 node010 ... node016
```

4. Скрипт catch_node.sh, функции которого состоят в мониторинге очереди заданий кластера, а также в создании start/stop-флагов для запускаемых/прерываемых MPI-задач.

5. Скрипт start_boinc.sh, функции которого заключаются в запуске и остановке работы MPI-задач start_boinc.

Далее мы кратко описываем работу СИСР CluBORun. Все действия осуществляются под некоторым именем пользователя, который имеет стандартный доступ к ресурсам кластера, т.е. может ставить свои MPI-задачи в очередь заданий кластера. С помощью стандартного планировщика cron устанавливается правило – запускать скрипт catch_node.sh через небольшие временные промежутки (например, каждую минуту). Данный скрипт анализирует очередь кластера, используя стандартные команды, доступные любому пользователю кластера. Конкретнее, скрипт catch_node.sh запрашивает список всех задач, запущенных на кластере, а также список задач, стоящих в очереди (если таковые есть). Анализируя эту информацию, скрипт определяет, сколько BOINC-задач уже запущено на кластере и сколько можно запустить еще, исходя из числа свободных узлов и состава очереди кластера. Результатом анализа является список MPI-задач start_boinc и список узлов кластера, на которых данные задачи могут быть запущены в настоящий момент. Всем таким задачам приписывается start-флаг. Аналогично, если catch_node.sh обнаруживает в очереди ожидающие обработки задачи сторонних пользователей, он принимает решение о прерывании некоторых

(возможно, всех) задач `start_boinc`. В каталоги соответствующих задач прописывается `stop`-флаг. Вся информация заносится в каталог соответствующей задачи `start_boinc` через обращение к файлу `all_tasks.txt`.

Скрипт `start_boinc.sh` периодически просматривает каталоги MPI-задач `start_boinc`. Если в каталоге имеется `start`-флаг, то данная задача запускается на соответствующем узле кластера стандартным способом (через вызов `mpirun`). Если же `start_boinc.sh` обнаруживает в каталоге некоторой задачи `stop`-флаг, то принимается решение о прерывании данной задачи. После получения от `start_boinc.sh` команды на прерывание работы управляющие MPI-процессы программы `start_boinc` выдают аналогичные команды запущенным на них BOINC-менеджерам. Последние сохраняют промежуточные результаты работы и освобождают ресурсы узлов кластера.

4. Вычислительные эксперименты

На текущем этапе CluBORun применялся в связке с проектами добровольных вычислений. Следует отметить, что сама по себе идея привлечения кластеров в проекты добровольных вычислений весьма плодотворна, поскольку ресурсы кластера надежнее ресурсов ПК как в плане предсказуемости времени решения соответствующих задач, так и в том, что получаемые на кластере результаты нет смысла фальсифицировать. Мы использовали CluBORun для усиления вычислительной мощности проектов добровольных вычислений SAT@home и OPTIMA@home.

Основная техническая трудность при подключении к CluBORun-грид новых кластеров заключается в том, что на различных кластерах используются различные системы управления очередью заданий. Соответственно, перечисленные выше алгоритмы взаимодействия с очередью приходится каждый раз переписывать с учетом возникающих нюансов. Первая версия CluBORun-грид [2] была создана для работы с системами управления очередями заданий кластеров Cleo [3] (разработка НИВЦ МГУ) и СУПЗ [4] (разработка МСЦ РАН). Эта версия CluBORun-грид была запущена на кластере Blackford ИДСТУ СО РАН [5] с 144 ядрами. Результаты работы CluBORun-грид на данном кластере с 12 апреля по 13 мая 2013 года представлены на рисунке 1. Синим цветом обозначены BOINC-расчеты, зеленым – расчеты, выполняемые другими пользователями кластера. Желтым цветом обозначены свободные ресурсы, красным – затребованные другими пользователями ресурсы в очереди заданий. Оранжевым цветом обозначены заблокированные ресурсы, то есть те ядра, доступ к которым был закрыт политикой администрирования. В данных экспериментах кластер был использован в связке с проектом добровольных вычислений SAT@home [6, 7].

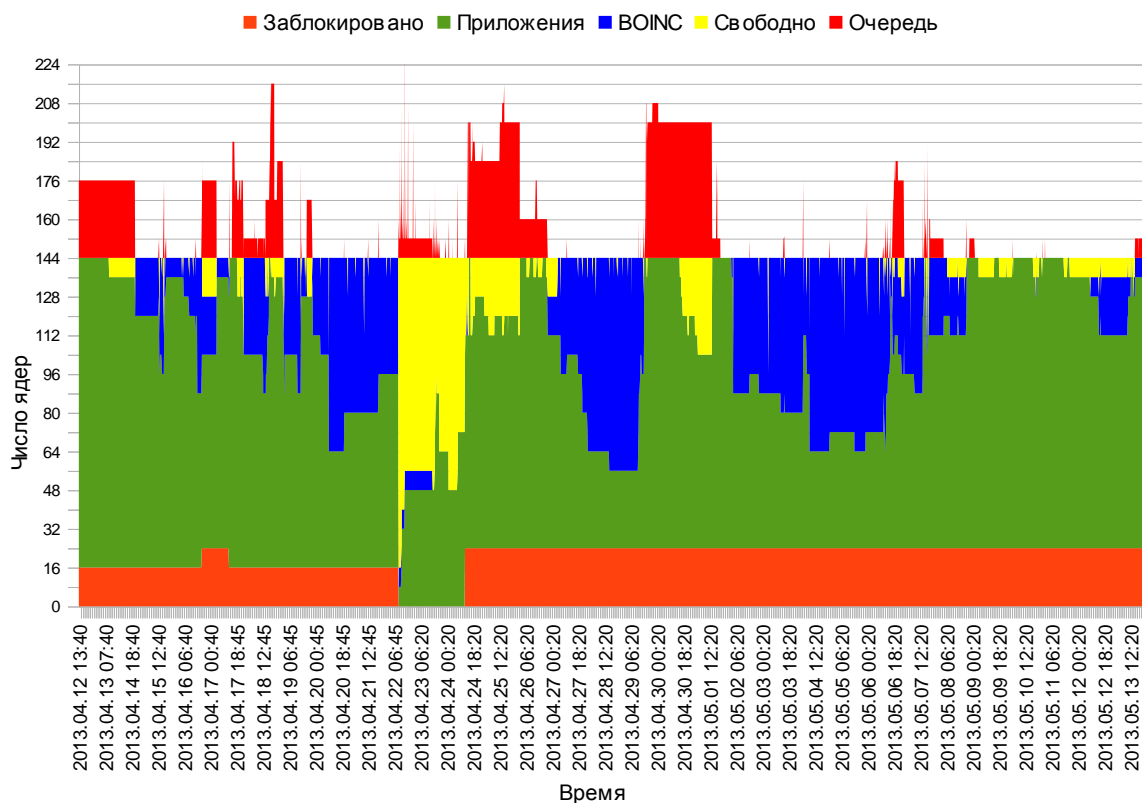


Рис. 1. Загрузка кластера Blackford ИДСТУ СО РАН с 12.04.2013 по 13.05.2013.

С 22 апреля по 27 апреля 2013 года на работу CluBORun накладывались жесткие ограничения по числу свободных узлов, из-за этого более половины ресурсов кластера простаивало. В следующий промежуток времени эти ограничения либо смягчались, либо снимались совсем, вследствие чего все незаблокированные узлы кластера были задействованы либо в обычных расчетах, либо в BOINC-расчетах. Также на рисунке 1 хорошо видно, как CluBORun освобождает от BOINC-расчетов узлы при появлении в очереди заданий других пользователей.

Отметим здесь, что на данной стадии экспериментов мы использовали следующую простейшую схему освобождения ресурсов: при появлении в очереди новых задач сторонних пользователей останавливалось такое количество BOINC-расчетов, которое было необходимо для запуска этих задач.

В декабре 2013 года CluBORun-грид была запущена на кластере MBC-100k [8], на котором установлена система СУПЗ. Специфика работы с MBC-100k привела сначала к появлению второй, а потом и третьей версии CluBORun-грид. Основное достижение этих версий состоит в новых алгоритмах обработки информации об очереди заданий кластера. Данные алгоритмы подробно описаны в [9].

Версия CluBORun-грид, развернутая на МВС-100к в настоящее время, позволяет использовать значительную часть ядер, не занятых обычными расчетными задачами. Так, с 1 по 20 апреля 2014 года в среднем на кластере было свободно 545 ядер, из которых примерно 237 ядер были задействованы для выполнения вычислительных задач SAT@home с помощью CluBORun. Результаты работы CluBORun на кластере МВС-100к с 5 по 12 марта 2014 года представлены на рисунке 2.

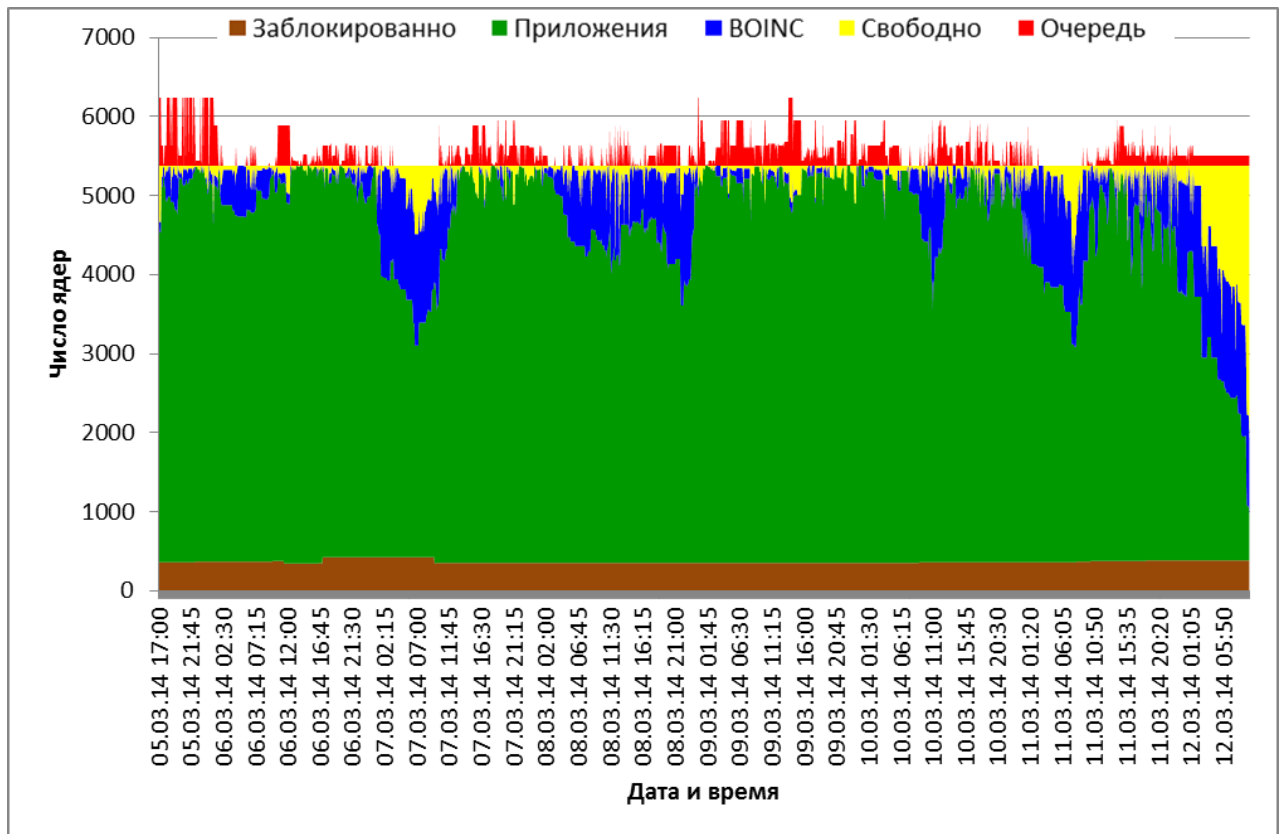


Рис. 2. Загрузка кластера МВС-100к с 05.03.2014 по 12.03.2014.

Программный комплекс CluBORun был использован для подключения кластеров к решению задач в проекте добровольных вычислений SAT@home. С помощью CluBORun были найдены 17 новых пар ортогональных диагональных латинских квадратов порядка 10 [10] (до этого были известны 3 такие пары, опубликованные в статье [11]). Также в SAT@home были решены 3 ослабленные задачи криптоанализа генератора ключевого потока Vivium. Ослабление заключалось в подстановке известных значений последних 10 из 177 бит искомого заполнения регистров генератора Vivium. Отметим, что в [12] рассматривалось решение на кластере ослабленных задач криптоанализа Vivium, но при 12 известных битах из 177. В некоторые периоды времени вклад задействованного с помощью CluBORun кластера МВС-100к в производительность SAT@home достигал 40 %.

Помимо SAT@home, CluBORun-грид используется в связке с проектом ДРВ ОПТИМА@home. Данный проект разработан специально для реализации

различных современных численных алгоритмов оптимизации. На текущий момент в OPTIMA@home решаются задачи минимизации функционалов, возникающих при молекулярно-динамическом моделировании свойств разнообразных материалов.

5. Сравнение CluBORun-грид с другими разработками.

При разработке CluBORun-грид мы ориентировались на идеи, используемые в программном инструментарии VNB-Grid [13]. Данный инструментарий предназначен для создания сервисных грид на основе вычислительных кластеров. В рамках VNB-Grid создаются MPI-задания, которые запускаются через систему управления очередями заданий кластеров (т.е. используются только обычные права пользователя кластера). Основную новизну результатов, описанных в настоящей статье, составляет автоматическая система мониторинга и использования свободных ресурсов кластера. Данная система, анализируя состояние очереди, планирует выделение ресурсов кластера под задачи, решаемые в CluBORun-грид.

Следует отметить также программный инструментарий 3G Bridge [14], предназначенный для обмена заданиями между разными грид-системами. Данный инструментарий может быть использован, например, для отправки заданий из BOINC-ориентированного грид из ПК в сервисной грид на основе вычислительных кластеров. Основное отличие 3G Bridge от CluBORun-грид состоит в том, что 3G Bridge ориентирован на использование уже готовых сервисных грид, а CluBORun может быть использован при доступе к кластерам с правами обычного пользователя. Кроме этого, в отличие от CluBORun-грид, в 3G Bridge не реализованы механизмы гибкого выделения свободных ресурсов кластеров.

6. Заключение и направления дальнейшей работы.

В настоящей статье мы описали концепцию многозадачной грид-системы с гибким управлением свободными ресурсами суперкомпьютеров. Нам представляется, что такая система могла бы использоваться для решения двух важных проблем распределенных вычислений: решение сверхмасштабных вычислительных задач и использование свободных ресурсов современных суперкомпьютеров, которые испытывают систематическую недогрузку. Мы описали первый реально работающий прототип подобной грид-системы – CluBORun-грид. Эта система на настоящий момент используется для увеличения вычислительной мощности проектов добровольных распределенных вычислений SAT@home и OPTIMA@home. Основной компонент новизны CluBORun-грид – автоматическая система мониторинга очереди и задействования свободных вычислительных ресурсов кластера.

В ближайшее время мы планируем добавить возможность работы с другими системами управления очередями кластеров, в частности, с PBS TORQUE. Исходный код CluBORun выложен в открытый доступ [15].

Отметим, что предложенный в настоящей статье подход может быть применен не только для добавления вычислительных ресурсов кластеров в проекты ДРВ. Перспективным представляется создание сервисных грид-систем, ориентированных на использование свободных ресурсов вычислительных кластеров, входящих в эти системы. Для управления такой грид-системой может быть задействован стандартный BOINC-сервер без функции подключения к нему частных лиц.

Нам представляется, что в более далекой перспективе описанные в статье технологии могут быть использованы для создания многозадачной грид-системы с широким спектром решаемых задач и с привлечением свободных ресурсов большого числа географически удаленных друг от друга суперкомпьютеров.

СПИСОК ЛИТЕРАТУРЫ

1. Anderson, D.P. BOINC: A System for Public-Resource Computing and Storage // In: Buyya, R. (ed.) GRID. pp. 4-10. IEEE Computer Society, 2004.
2. Манзюк М.О., Заикин О.С. CluBORun: средство использования свободных ресурсов вычислительных кластеров для BOINC-расчетов // Труды научной конференции «Высокопроизводительные вычисления на базе BOINC: фундаментальные исследования и разработки». ИПМИ КарНЦ РАН. 2013. С. 9-14.
3. Система управления заданиями Cleo. URL: <http://parcon.parallel.ru/cleo.html> (дата обращения: 6.10.2014).
4. Система управления прохождением задач (СУПЗ). URL: <http://suppz.jssc.ru/> (дата обращения: 6.10.2014).
5. Суперкомпьютерный центр Иркутского научного центра СО РАН. URL: <http://hpc.icc.ru> (дата обращения: 6.10.2014).
6. SAT@home: проект добровольных вычислений для решения трудных SAT-задач. URL: <http://sat.isa.ru/pdsat/> (дата обращения: 6.10.2014).
7. Заикин О.С., Посыпкин М.А., Семенов А.А., Храпов Н.П. Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home // Вестник ННГУ. № 5(2). 2012. С. 338-346.
8. Суперкомпьютерный центр РАН. URL: <http://www.jssc.ru/scomputers.shtml> (дата обращения: 6.10.2014).
9. Манзюк М.О., Заикин О.С., Посыпкин М.А. CluBORun: программный комплекс для использования свободных ресурсов вычислительных кластеров в BOINC-расчетах // Информационные технологии и вычислительные системы (в печати).

10. Заикин О. С., Кочемазов С.Е., Семенов А.А. Поиск систем ортогональных латинских квадратов в проекте добровольных вычислений SAT@home // Труды первой российской конференции «Высокопроизводительные вычисления на базе BOINC: фундаментальные исследования и разработки». ИПМИ КарНЦ РАН. 2013. С. 3-8.
11. Brown et al. Completion of the Spectrum of Orthogonal Diagonal Latin Squares. Lecture notes in pure and applied mathematics. 1992. Vol. 139. pp. 43–49.
12. Заикин О.С., Семенов А.А. Применение метода Монте-Карло к прогнозированию времени параллельного решения проблемы булевой выполнимости // Вычислительные методы и программирование: Новые вычислительные технологии. 2014. Вып. 1. С. 22-35.
13. Alexander Semenov, Oleg Zaikin, Dmitry Bepalov, Mikhail Posypkin. Parallel Logical Cryptanalysis of the Generator A5/1 in BNB-Grid System // Lecture Notes in Computer Science. Vol. 6873, 2011, pp 473-483.
14. Zoltan Farkas, Peter Kacsuk, Zoltan Balaton, Gabor Gombas. Interoperability of BOINC and EGEE // Future Generation Computer Systems. Vol. 26, no. 8, pp. 1092-1103, 2010.
15. Программные комплекс CluBORun. URL: <https://github.com/Nauchnik/CluBORun> (дата обращения: 6.10.2014).